

L16 ANSWER 7 OF 7 USPATFULL  
 AN 93:66109 USPATFULL  
 TI Access control subsystem and method for distributed computer system  
 using locally cached authentication credentials  
 IN Wobber, Edward, Menlo Park, CA, United States  
 Abadi, Martin, Palo Alto, CA, United States  
 Birrell, Andrew, Los Altos, CA, United States  
 Lampson, Butler, Cambridge, MA, United States  
 PA Digital Equipment Corporation, Maynard, MA, United States (U.S.  
 corporation)  
 PI US 5235642 19930810  
 AI US 1992-917767 19920721 (7)  
 DT Utility  
 FS Granted  
 LN.CNT 604  
 INCL INCLM: 380/025.000  
 INCLS: 380/004.000  
 NCL NCLM: 713/156.000  
 NCLS: 713/158.000; 713/164.000  
 IC [5]  
 ICM: H04K001-00  
 EXF 380/23; 380/25; 380/4  
 SUMM A further optimization is that the server process local cache is used to  
 store a list of the object access control list entries previously  
 satisfied by each requester, thereby enabling the **server**  
 process to expedite granting access to **previously**  
**accessed** objects.  
  
 DETD Returning to step 206, if the requester is listed in the server's local  
 cache 164, and the timestamp for the requester indicates that the  
 previously received credentials for this requester are still valid, the  
 server process proceeds with execution of the requested tasks (step  
 218). During execution of these tasks, if the server process  
 successfully gains access to any objects on behalf of the requester, the  
 ACL entries satisfied by the requester are added by the server process  
 to the requester's record in the server process's local cache (step  
 220). The storage of ACL entries known to be satisfied by a particular  
 requester in the server's local cache can be used by the **server**  
 process to expedite granting access to **previously**  
**accessed** objects.  
 NCL NCLM: 713/156.000  
 NCLS: 713/158.000; 713/164.000

L16 ANSWER 3 OF 7 USPATFULL  
 AN 1999:97793 USPATFULL  
 TI Information delivery system and method including restriction processing  
 IN Zucknovich, Stephen M., Wayne, NJ, United States  
 Leisy, Jacques, Bridgewater, NJ, United States  
 Kitain, Eduard, Brooklyn, NY, United States  
 Urazov, Yuri, Forest Hills, NY, United States  
 Baird, George, New York, NY, United States  
 Blazek, Paul, Forest Hills, NY, United States  
 Prohorov, Dmitry, Forest Hills, NY, United States  
 Kolfman, Michael, Brooklyn, NY, United States  
 Yackubovich, Alex, Highland Park, NJ, United States  
 PA Multex Systems, Inc., New York, NY, United States (U.S. corporation)  
 PI US 5940843 19990817  
 AI US 1997-947257 19971008 (8)  
 DT Utility  
 FS Granted  
 LN.CNT 2550  
 INCL INCLM: 707/516.000  
 INCLS: 707/002.000; 707/009.000; 707/010.000; 705/035.000; 395/188.010;  
 395/200.490  
 NCL NCLM: 707/516.000  
 NCLS: 705/035.000; 707/002.000; 707/009.000; 707/010.000; 709/219.000;  
 713/202.000  
 IC [6]  
 ICM: G06F017-21  
 EXF 707/9; 707/10; 707/516; 707/2; 705/35; 395/200.49; 395/188.01  
 DETD The contributor of a report can be notified that a particular investor  
 has **accessed** that report. The repository **server 2**  
 maintains for each report a list of those who **accessed** that  
 report. The repository **server 2** can transmit that list to the  
 report's contributor on a regular basis and/or when requested by the  
 contributor.  
 DETD The repository server 2 is coupled to a web server 4 which in turn is  
 coupled to the Internet via, for example, a T1 or ISDN connection. The  
 web server 4 is a high powered server computer that runs a web server  
 program. In the representative embodiment, the web server 4 executes,  
 for example, Netscape's Commerce Server program. The web server program  
 allows web pages (in HTML format) to be **accessed** by investors.  
 The web **server 4** also executes other programs and subroutines  
 as required.  
 DETD c. If the value is not empty, the CGI program indicates that this user  
 has **previously** already **accessed** the web  
**server 4** since starting the browser program, and has been given  
 an authorizing cookie. If the "mxauth" value of the cookie does not  
 match the value stored on the web server for this user, then this user  
 has been superseded by another user using the same ID. The CGI does not  
 perform the requested task, and tells the user that access is denied. If  
 the "mxauth" value of the cookie does match, then this user is  
 authorized to continue, and the CGI performs the requested task. Each  
 time the user is authorized to continue, the time of the access is  
 stored on the web server 4.  
 NCL NCLM: 707/516.000  
 NCLS: 705/035.000; 707/002.000; 707/009.000; 707/010.000; 709/219.000;  
 713/202.000

L16 ANSWER 2 OF 7 USPATFULL  
 AN 1999:114760 USPATFULL  
 TI Method and apparatus for protecting data files on a computer from virus infection  
 IN Walsh, James E., Kirkland, WA, United States  
 Altberg, Ebbe H. A., Bellevue, WA, United States  
 PA Microsoft Corporation, Redmond, WA, United States (U.S. corporation)  
 PI US 5956481 19990921  
 AI US 1997-797485 19970206 (8)  
 DT Utility  
 FS Granted  
 LN.CNT 1275  
 INCL INCLM: 395/186.000  
 INCLS: 380/004.000  
 NCL NCLM: **713/200.000**  
 NCLS: **713/188.000**  
 IC {6}  
 ICM: G06F012-16  
 EXF 395/183.14; 395/183.15; 395/183.12; 395/186; 395/682; 395/680; 364/580; 380/4  
 SUMM A utility program typically scans local files in response to booting the computer or during a predetermined time period for operation of a computer. Alternatively, if you access a file on a local machine, the utility program can scan the file at that time. Because utility programs typically offer virus protection by scanning files residing on a local machine, these utility programs can fail to address certain file events that may arise in a computer network environment, such as accessing a file on a remote server. For example, a utility program cannot scan a file that resides outside of the local user's machine, such as a file **accessed** via a remote **server**.  
 CLM What is claimed is:  
 22. A computer-implemented method for protecting a plurality of files on a computer from infection by a known virus component using a virus check routine incorporated within a program module, the program module operative to access the files and the virus check routine operative to store a digital signature with a selected data file once the selected data file is **accessed** by the program module, comprising the steps of: detecting a request to access the selected data file in response to one of an external and internal open file event; determining whether the selected data file contains the known virus component; if the selected data file contains the known virus component, then determining whether the selected data file was **previously accessed** by the program module by (i) obtaining the digital signature for the selected data file; (ii) obtaining a digital session key for the present session of the program module; and (iii) comparing the digital signature with the digital session key; if the digital signature matches the digital session key, then determining that the selected data file was **previously accessed** by the program module; determining whether the selected data file was **previously accessed** using a safe access mode; and if the selected data file was **previously accessed** using the safe access mode, then accessing the selected data file using the safe access mode.  
 NCL NCLM: **713/200.000**  
 NCLS: **713/188.000**